

Zadanie1. Kwadrat

Napisz aplikację WPF w języku C#, która pozwala użytkownikowi obliczać pole i obwód kwadratu na podstawie podanej przez niego długości boku.

Aplikacja powinna zawierać dwa przyciski: jeden do obliczania pola, a drugi do obliczania obwodu kwadratu. Wyniki obliczeń powinny być wyświetlane w odpowiednim polu tekstowym.

Szczegóły implementacji:

1. Interfejs użytkownika (XAML):

- TextBox do wprowadzania długości boku kwadratu.
- Dwa przyciski: jeden do obliczania pola, drugi do obliczania obwodu kwadratu.
- TextBlock do wyświetlania wyników.

2. Logika w kodzie C#:

- Przyciski powinny mieć zdefiniowane zdarzenia kliknięcia (Click), które uruchamiają odpowiednie metody.
- Metody te powinny pobierać wartość z TextBox, sprawdzać, czy jest ona liczbą całkowitą, i obliczać odpowiednio pole lub obwód kwadratu.
- Jeśli wprowadzona wartość nie jest liczbą, użytkownik powinien otrzymać komunikat o błędzie.



KWADRAT

Podaj długość boku:

22 * 22 = 484

licz pole

licz obwód



KWADRAT

Podaj długość boku:

4 * 22 = 88

licz pole

licz obwód

KWADRAT

Podaj długość boku:

Należy wpisać wartość liczbową

licz pole

licz obwód

Zadanie 2. Kalkulator

Napisz aplikację WPF (Windows Presentation Foundation) w języku C#, która pozwala użytkownikowi wykonywać podstawowe operacje arytmetyczne **na dwóch liczbach całkowitych**: dodawanie, odejmowanie, mnożenie i dzielenie.

Aplikacja powinna zawierać pola tekstowe do wprowadzania dwóch liczb, przyciski do wyboru operacji oraz pole tekstowe do wyświetlania wyników.

Szczegóły implementacji:

1. Interfejs użytkownika (XAML):

- Dwa TextBoxy do wprowadzania liczb (a i b).
- Cztery przyciski do wykonywania operacji: dodawanie, odejmowanie, mnożenie i dzielenie.
- TextBlock do wyświetlania wyników i komunikatów.

2. Logika w kodzie C#:

- Każdy przycisk powinien mieć zdefiniowane zdarzenie kliknięcia (Click), które uruchamia odpowiednie metody do wykonania operacji arytmetycznych.
- Metody te powinny pobierać wartości z TextBoxów, sprawdzać, czy są to liczby całkowite, a następnie wykonywać odpowiednią operację arytmetyczną.
- Jeśli wprowadzone wartości nie są liczbami całkowitymi lub jedno z pól jest puste, użytkownik powinien otrzymać komunikat o błędzie.
- Dodatkowo, przy operacji dzielenia, należy obsłużyć przypadek dzielenia przez zero.

MainWindow

Podaj pierwszą liczbę

Podaj drugą liczbę

4 - 5 = -1

DODAWANIE ODEJMOWANIE MNOŻENIE DZIELENIE

MainWindow

Podaj pierwszą liczbę

Podaj drugą liczbę

Proszę uzupełnić obie liczby

DODAWANIE ODEJMOWANIE MNOŻENIE DZIELENIE

MainWindow

Podaj pierwszą liczbę

Podaj drugą liczbę

Nie wolno dzielić przez 0

DODAWANIE ODEJMOWANIE MNOŻENIE DZIELENIE

Zadanie 3 : Obliczanie wynagrodzenia za przepracowane godziny

Stwórz aplikację desktopową przy użyciu technologii WPF, która umożliwi obliczenie wynagrodzenia na podstawie liczby przepracowanych godzin i stawki godzinowej.

Interfejs użytkownika:

1. Użytkownik wprowadza liczbę przepracowanych godzin w polu tekstowym.
2. Użytkownik wprowadza stawkę godzinową w polu tekstowym.

Funkcjonalność:

- Po kliknięciu przycisku "Oblicz wynagrodzenie" aplikacja ma obliczyć wynagrodzenie na podstawie wprowadzonych danych.

- Obliczone wynagrodzenie powinno zostać wyświetlone za pomocą okna modalnego MessageBox.

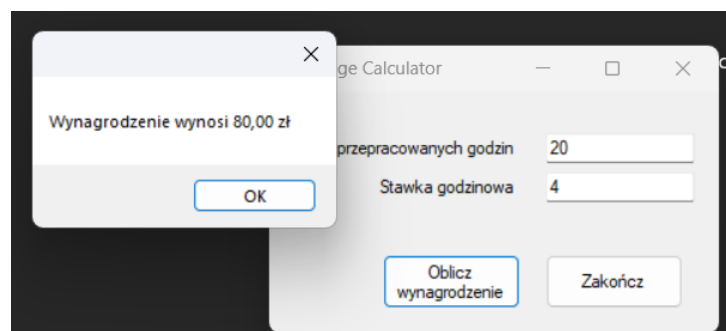
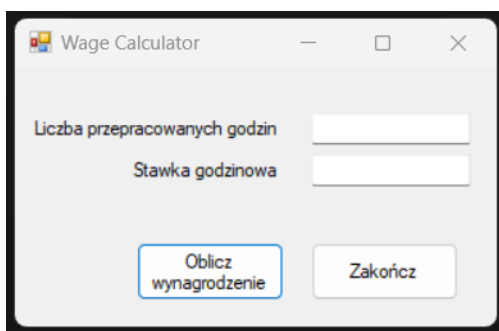
Warunki:

- Aplikacja powinna obsługiwać sytuacje, gdy użytkownik wprowadzi nieprawidłowe dane (np. ujemną liczbę godzin lub ujemną stawkę godzinową) poprzez odpowiednie komunikaty błędów w oknie modalnym.
- Wynagrodzenie powinno być obliczane jako iloczyn liczby przepracowanych godzin i stawki godzinowej.

Przykład:

- Użytkownik wprowadza liczbę godzin: 40
- Użytkownik wprowadza stawkę godzinową: 20 zł/h
- Po kliknięciu przycisku "Oblicz", aplikacja wyświetla wynagrodzenie: 800 zł

Rys. poglądowy :



Zadanie 4. Remont mieszkania.

Stwórz aplikację desktopową w języku C# z wykorzystaniem technologii WPF, która będzie obliczać powierzchnię ścian pokoju oraz koszt ich malowania.

Wymagania

1. Interfejs użytkownika:

- Dwa pola tekstowe do wprowadzania szerokości i długości pokoju.
- Przycisk, który po kliknięciu uruchomi obliczenia.
- Dwa pola tekstowe lub etykiety do wyświetlania wyników:
 - Powierzchni ścian.
 - Kosztu malowania.

2. Logika aplikacji:

- Po kliknięciu przycisku, aplikacja powinna odczytać wartości wprowadzone przez użytkownika (szerokość i długość pokoju).
- Sprawdź, czy wprowadzone wartości są liczbami zmiennoprzecinkowymi. Jeśli nie wyświetl komunikat o błędzie.
- Sprawdź, czy wprowadzone wartości są liczbami dodatnimi. Jeśli którakolwiek z wartości jest mniejsza lub równa zero, wyświetl komunikat o błędzie.
- Jeśli wprowadzone wartości są poprawne, oblicz powierzchnię ścian pokoju przy założeniu, że wysokość pokoju wynosi 2.7 metra:
- Oblicz koszt malowania przy założeniu, że koszt malowania wynosi 8 zł za metr kwadratowy.
- Wyświetl obliczone wartości (powierzchnię ścian i koszt malowania) na ekranie.

Szczegółowe wytyczne GUI:

Ogólne informacje

- Nazwa okna: Remont
- Wymiary okna: 800 szerokości, 450 wysokości

- Tło okna: Jasnoniebieskie (LightBlue)

Elementy interfejsu użytkownika

1. Nagłówek aplikacji

- Element: Label
- Treść: "Malowanie pokoju: 8 zł za m²"
- Położenie: W lewym górnym rogu
- Wymiary: 618 szerokości, 77 wysokości
- Marginesy: 37 od lewej, 34 od góry
- Styl: Czcionka o rozmiarze 30, pogrubiona

2. Podtytuł / Opis

- Element: Label
- Treść: "Wymiary pokoju w metrach:"
- Położenie: W lewym górnym rogu, poniżej nagłówka
- Wymiary: 189 szerokości, 46 wysokości
- Marginesy: 37 od lewej, 130 od góry
- Styl: Czcionka o rozmiarze 14

3. Pole tekstowe do wprowadzania szerokości pokoju

- Element: TextBox
- Położenie: Na lewo od środka, poniżej opisu
- Wymiary: 150 szerokości, 23 wysokości
- Marginesy: 262 od lewej, 130 od góry
- Inne: zawijanie tekstu

4. Pole tekstowe do wprowadzania długości pokoju

- Element: TextBox
- Położenie: Na prawo od środka, poniżej opisu
- Wymiary: 150 szerokości, 23 wysokości

- Marginesy: 440 od lewej, 130 od góry
- Inne: zawijanie tekstu

5. Przycisk do obliczeń

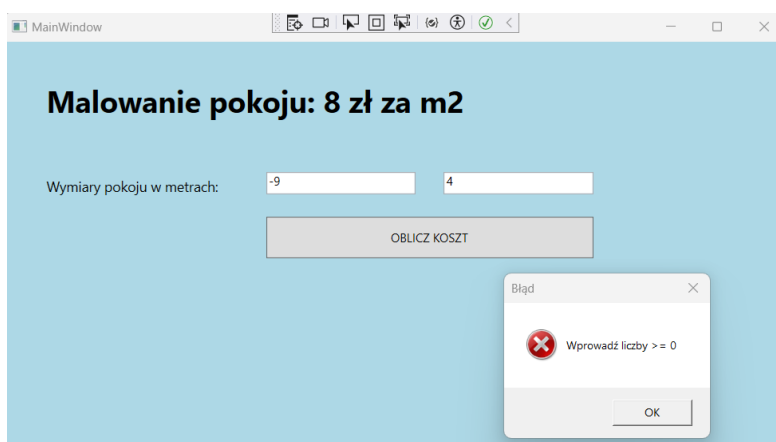
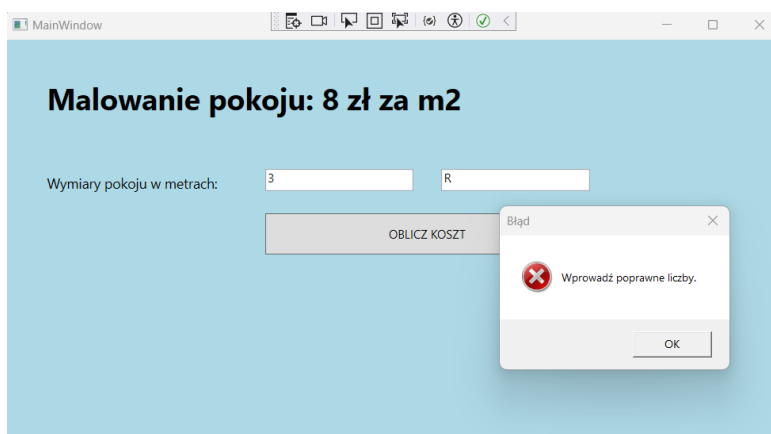
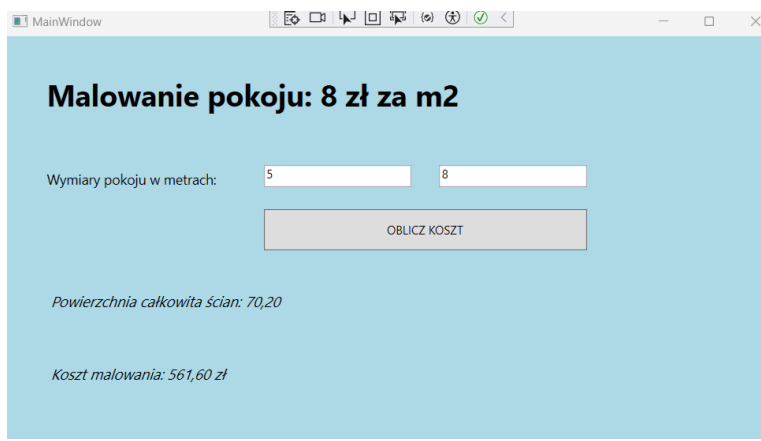
- Element: Button
- Treść: "OBLICZ KOSZT"
- Położenie: Na środku, poniżej pól tekstowych
- Wymiary: 328 szerokości, 42 wysokości
- Marginesy: 262 od lewej, 175 od góry

6. Etykieta do wyświetlania powierzchni ścian

- Element: Label
- Treść: Pusty początkowo
- Położenie: W lewym górnym rogu, poniżej przycisku
- Wymiary: 657 szerokości, 39 wysokości
- Marginesy: 41 od lewej, 253 od góry
- Styl: Czcionka o rozmiarze 15, styl kursywa

7. Etykieta do wyświetlania kosztu malowania

- Element: Label
- Treść: Pusty początkowo
- Położenie: W lewym górnym rogu, poniżej etykiety z powierzchnią
- Wymiary: 657 szerokości, 36 wysokości
- Marginesy: 41 od lewej, 327 od góry
- Styl: Czcionka o rozmiarze 15, styl kursywa

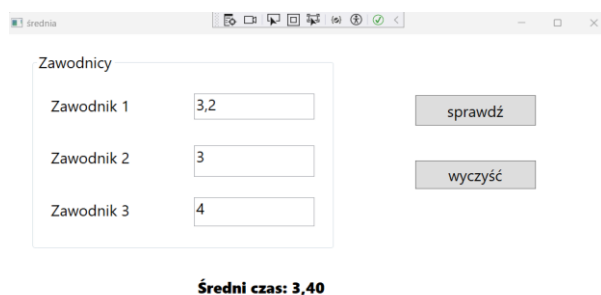


Zadanie 5. Średni czas zawodników.

Zaimplementuj aplikację okienkową w języku C# (WPF) "Średnia czasu zawodników", która umożliwia wprowadzenie czasów trzech zawodników oraz obliczenie ich średniego czasu. Zastosuj kontrolę wprowadzanych wartości.

1. Stwórz okno główne o tytule "Średnia czasu zawodników" o wymiarach 800x450 pikseli i rozmiarze czcionki ustawionym na 20 punktów.
2. Umieść na oknie głównym siatkę (**Grid**), która będzie zawierała następujące elementy:
 - a) Grupę elementów (**GroupBox**) o nazwie "Zawodnicy", umieszczoną z lewej strony, zawierającą trzy etykiety (**Label**) i trzy pola tekstowe (**TextBox**) odpowiednio dla każdego z trzech zawodników.
 - b) Etykiety (**Label**) dla każdego z trzech zawodników (np. "Zawodnik 1", "Zawodnik 2", "Zawodnik 3") umieszczone obok pól tekstowych (**TextBox**), aby wprowadzić czasy dla każdego zawodnika.
 - c) Przycisk "sprawdź" (**Button**), umieszczony z prawej strony, który będzie uruchamiał obliczenie średniego czasu na podstawie wprowadzonych danych.
 - d) Przycisk "wyczyść" (**Button**), umieszczony poniżej przycisku "sprawdź", służący do wyczyszczenia wprowadzonych danych (implementacja brakuje).
 - e) Etykiety (**Label**) do wyświetlenia wyniku obliczonej średniej czasu, umieszczone poniżej przycisków.
3. Dodaj obsługę zdarzenia dla przycisku "sprawdź", aby obliczyć średni czas na podstawie wprowadzonych danych, a następnie wyświetlić wynik za pomocą etykiety (**Label**).
4. Zadbaj o odpowiednie rozmieszczenie elementów w oknie głównym, aby interfejs użytkownika był czytelny i estetyczny.

Rys.1



Rys.2

